PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-331355
(43)Date of publication of application : 30.11.2001

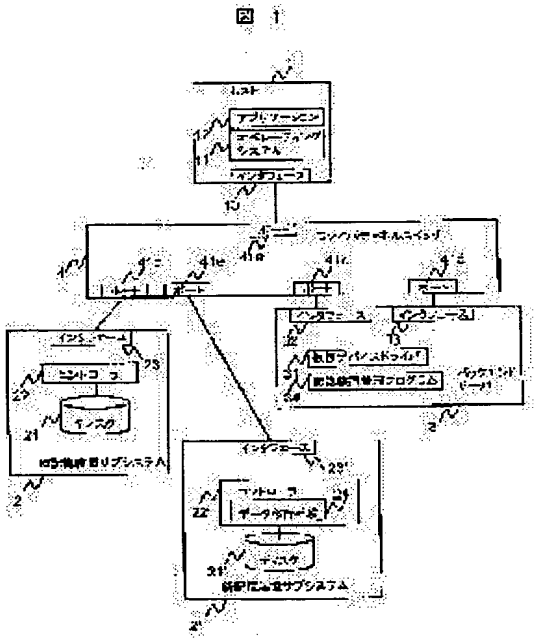
(51)Int.Cl. G06F 12/00
G06F 3/06

(21)Application number : 2000-152672
(22)Date of filing : 18.05.2000

(71)Applicant : HITACHI LTD
(72)Inventor : KITAMURA MANABU
ARAI HIROHARU

(54) COMPUTER SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To transmissively execute data transition among storage devices to host computers in a computer system, in which plural host computers are connected with plural storage devices.
SOLUTION: A back-end server 3 provides a host 1 having a virtual disk. The virtual disk first appears to be same as an old storage device sub-system 2 to the host 1. When data is transited from the old storage device sub-system 2 to a new storage device sub-system 2, the back-end server 3 first instructs a data transition processing to the new storage device sub-system 2, and after that, switches setting of the virtual disk and makes it correspond to the new storage device sub-system 2. Data transition among the disk devices is performed transmissively to the host 1, since this switching is transmissively executed to the host 1.



DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention]This invention relates to the data accessing method of the storage system in an information processing system etc., and relates to the data transition method in memory storage especially.
[0002]
[Description of the Prior Art]The architecture from which a personal computer, a workstation, a mainframe, etc. differ, the plat form which has adopted the operating system, and two or more memory storage are connected mutually, and the motion summarized to what is called one network prospers. Generally this is called SAN (Storage AreaNetwork) in the language to LAN (Local Area Network) which connected two or more computers in networks, such as Ethernet (registered trademark) (Ethernet (registered trademark)). SAN usually connects memory storage with a computer using the transmission line of an optical cable called a fiber channel (Fibre Channel) thru/or copper wire.
[0003]Some advantages are raised to SAN. It is providing the 1st with the environment which memory storage can access in common from two or more computers first. The data transfer between memory storage is possible by carrying out interconnection of the memory storage to the 2nd, the data copy between backup or memory storage can be realized by this, without applying load to a host computer, and the change to the memory storage of a sub system is attained at the time of a memory storage obstacle. Since each memory storage was connected [3rd] to the computer of former each, management (surveillance of the state of a device, change of setting out) of memory storage enables management of all the memory storage for what was made only from each computer connected from a specific computer. Although a maximum of 16 sets only of apparatus have been connected in the conventional SCSI (Small Computer System Interface), by a fiber channel, 100 or more sets of apparatus can be connected on-line, and easy extendibility can be obtained.
[0004]There is nothing that actually employed the above-mentioned advantage efficiently, although many products for realizing SAN in recent years are appearing. In especially extendibility, although the online connection of apparatus became possible physically, the base art of utilizing it is insufficient. For example, in SAN, when a disk unit is newly extended for exchange of a disk unit, extension of apparatus can be carried out on-line, but a user needs to move data clearly after it. In order for a user to enjoy a merit by on-line apparatus extension, data movement etc. need to be carried out transparent to a user not only with extension of simple hardware but with extension of hardware.
[0005]The example is indicated by the U.S. Pat. No. 5680640 item about movement of the on-line data between disk units. Although a U.S. Pat. No. 5680640 item is the data transition on condition of the disk for mainframes, the communication wire which connects between disk units is used, short-time cutting of the connection between disk units is only carried out with a host, and the rest makes data transition between disk units possible transparent to a user.
[0006]
[Problem(s) to be Solved by the Invention]The U.S. Pat. No. 5680640 item makes data movement

between disk units possible transparent infinite to the user. However, this is the data transition method on condition of the disk for mainframes, and application in SAN cannot be performed. In a U.S. Pat. No. 5680640 item, when changing the old disk unit to a new disk unit, it can pretend to be the host side by setting out by the side of a disk unit, as if the new disk was the old disk unit. This is possible by operating setting out of the device number of a disk unit, etc.
[0007]However, meaning ID given to each disk is determined by the negotiation of the apparatus (a disk unit, a fiber channel switch) which constitute a network, and is not changed into the case of SAN, for example, fiber channel environment, by a user's setting out. When using the U.S. Pat. No. 5680640 No. data transition method, to a host computer, it cannot make a show of a new disk unit as an old disk unit, and data transition transparent to a user as a matter of fact cannot be realized. To a host and a user, the purpose of this invention is transparent and there is in providing the system which can employ the extendibility of SAN efficiently.
[0008]
[Means for Solving the Problem]A computer system in this invention comprises a switch which connects a host computer, a back end computer, two or more storage subsystems, and a host computer and a back end computer. Although a host computer accesses each storage subsystem via a back end computer, a back end computer provides one thru/or two or more virtual disk units to a host computer. If a virtual disk unit has an access request from a host computer, by a back end computer, a demand will be suitably given to a storage subsystem actually connected according to a kind of virtual disk unit with a demand.
[0009]
[Embodiment of the Invention]Drawing 1 is a block diagram showing the example of composition in one embodiment of the computer system which applied this invention. A computer system comprises the host 1, the old storage subsystem 2, the new storage subsystem 2, the back end server 3, and the fiber channel switch 4.
[0010]The host 1 comprises the operating system 11, the application 12, and the interface 13. Although it operates on CPU on the host 1, and a memory actually, the operating system 11 and the application 12 are omitted in order not to have the contents and the relation of this invention about the component of these hardwares. Although the environment where two or more host computers are actually connected in addition to host 1 is common, since it is easy, by this invention, only the host 1 has been indicated as a host computer. The old storage subsystem 2 comprises the disk 21, the controller 22, and the interface 23. Even if the disk 21 is the logical drive which packed two or more physical disks and was made to look like one logical disk unit, there is no change in the contents of this invention. The interface 23 is connected with the fiber channel switch 4. The new storage subsystem 2 as well as the old storage subsystem 2 comprises the disk 21, the controller 22, and the interface 23. A different point from the old storage subsystem 2 is that the data transition means 24 is contained in the controller 22.
[0011]The back end server 3 comprises the virtual device driver 31 and the interfaces 32 and 33. Although the virtual device driver 31 is the software which operates on CPU on the back end server 3, and a memory and it is possible to change setting out from the exterior by a user, or to replace the program itself, About hardware components, such as CPU and a memory, since it is not related to the contents of this invention, it is omitting.
[0012]The fiber channel switch 4 comprises two or more ports 41a, 41b, 41c, 41d, and 41e (it names generically below and abbreviates to the port 41), and it is used in order to connect the host 1, the old storage subsystem 2, the new storage subsystem 2, and the back end server 3 mutually. From [each] the port 41a, it is possible to access the ports 41b, 41c, 41d, and 41e. Therefore, although the host 1 can also access the direct old storage subsystem 2 and the new storage subsystem 2 from the ports 41b and 41e, In this embodiment, the host 1 decides to access the storage subsystem 2 via the back end server 3 altogether fundamentally.
[0013]The role of the back end server 3 is explained. With the virtual device driver 31, the back end server 3 is seen from the host 1, and it is visible as if they were one thru/or two or more disk units. According to this embodiment, if the host 1 looks at the interface 33 via the port 41d, it shall seem that one disk is connected. Henceforth, this disk is called a virtual disk. At first, from the host 1, the virtual device driver 31 is set up so that a virtual disk may be visible to the same thing as the disk

21 of the old storage subsystem 2. Namely, if the host 1 accesses logic block-address [of a virtual disk] (LBA) 0, or LBA1, The virtual device driver 31 accesses LBA0 or LBA1 of the disk 21 via the interface 32 and the port 41c, and returns a result to the host 1 via the interface 33 and the port 41d. Although it is used in order that the interface 32 may access the disk 21 and the disk 21, and the interface 33 is used for the exchange with the host 1 in the embodiment of this invention, it is possible to also make one interface perform these two roles. It is possible to make it visible [a virtual disk] to the disk 21 of the new storage subsystem 2 by changing setting out of the virtual device driver 31. When a setting variation is performed, it is changeless to the virtual disk which appears from a host computer. In the case of a disk unit with a fiber channel interface, from a host computer, can recognize a disk unit uniquely by port ID and a Logical Unit Number (LUN), but. Even if setting out of a virtual device driver is changed and a virtual disk is changed into the disk 21 from the disk 21, the ports ID and LUN of the virtual disk which appears to the host 1 do not change, and the host 1 does not have recognition of the actually accessed disk having changed. Next, the data transition means 24 of the new storage subsystem 2 is explained. The data transition means 24 has the same means as what is indicated by the U.S. Pat. No. 5680640 item. If shift of data is directed, the data transition means 24 will read data sequentially from the head of the disk 21 of the storage subsystem 2, and will write data in the disk 21. If it has a table which furthermore records whether shift of data was completed per each block or two or more blocks and read access comes during transition processing, With reference to this table, data is read from the disk 21 about the field where data transition is not settled, and the data of the disk 21 is returned about the field where data transition has ended.

[0014]Drawing 2 expresses the table 100 which a data transition means has. By the data transition means 24, data is copied to the disk 21 from the disk 21 for every block. On the table 100, it has the flag 102 every address 101. It is shown that the data of the address was already copied to the disk 21 from the disk 21 when the flag 102 was 1, and, in the case of 0, it is un-copying. A certain thing is shown. This table 100 is used in data transition processing, and the lead under data transition processing and light processing.

[0015]Drawing 3 explains the flow of the transition processing which the data transition means 24 performs. The counter B is prepared first and an initial value is set to 0 (Step 2001). Next, with reference to the table 100, it is confirmed whether the flag 102 of LBA B is 1 (Step 2002). Since data transition has ended when a flag is 1, the counter B is increased one time (Step 2005). At Step 2002, if the flag 102 is 0, data will be copied to the disk 21 from the disk 21 (Step 2003), the flag 102 with which the table 100 corresponds will be updated to 1 (Step 2004), and it will progress to Step 2005. At Step 2006, it is confirmed whether it processed to the last LBA of the disk 21. That is, if it confirmed whether B exceeded the last LBA of the disk 21, it has exceeded and it completes and is not over processing, it returns to Step 2002, and processing is repeated.

[0016]Next, by drawing 4, while the data transition means is performing data transition processing of drawing 3, processing when the write request from the back end server 3 occurs is explained at a high order host, i.e., this embodiment. This processing updates to 1 the flag 102 of LBA which is easy and writes data in the disk 21 at Step 2101 and with which the table 100 corresponds at Step 2102. That is, data transition from the disk 21 is not performed about LBA to which light processing was performed during data transition processing.

[0017]By drawing 5, while the data transition means is performing data transition processing of drawing 3, processing when there is a lead demand from the back end server 3 is explained at a high order host, i.e., this embodiment. The flag 102 is referred to for LBA which had the lead demand in the table 100 at Step 2201. It confirms whether the flag 102 is 1 at Step 2202, and processing is branched. Since the data transition from the disk 21 is completed about the LBA when a flag is 1, data is read from the disk 21 (Step 2203). Since data transition is not completed about the LBA when a flag is 0, data is once copied to the disk 21 from the disk 21 (Step 2205). Then, the flag 102 of the table 100 is updated to 1 (Step 2206), and it progresses to henceforth [Step 2203]. The data read at Step 2204 is passed to the back end server 3, and processing is completed.

[0018]Next, the flow of the whole system is explained about the data transition processing to the new storage subsystem 2 in the system of this embodiment from the old storage subsystem 2. When performing data transition, a user directs shift to the back end server 3. Directions of the

data transition processing start from the back end server 3 to the new storage subsystem 2 are told to the new storage subsystem 2 via the interface 32. Drawing 6 explains the flow of processing of the back end server 3. The back end server 3 will suspend operation of the virtual disk by the virtual device driver 31 first, if directions of shift are received. (Step 1001). Thereby, access to the old storage subsystem 2 from the virtual device driver 31 is interrupted, and even if the virtual device driver 31 receives the access command to a virtual disk from the host 1, it does not return a response until an access stop is canceled. Next, the memory storage control program 34 directs the start of data transition processing to the new storage subsystem 2 (Step 1002). The data transition processing which the new storage subsystem 2 performs is mentioned later. Setting out of the virtual device which the virtual device driver 31 showed to the host 1 until now is changed so that access to the disk 21 may be performed, and access stopped by Step 1001 is made to resume at Step 1004 in Step 1003. When access of the virtual disk was resumed and access is coming from the host 1 to the virtual disk between Step 1001 and Step 1002, the access is altogether carried out to the disk 21.

[0019]In this embodiment, although the storage subsystem 2 and the back end server 3 were the topologies which led to the switch directly, even if it is the composition that the storage subsystem 2 is connected via the back end server 3 like drawing 7, realization is possible. When the storage subsystem 2 extended newly does not have the data transition means 24 like the example of drawing 1, the data transition means 24 is given to the back end server 3 side like drawing 8, and the same thing can be realized by making data transition processing perform by a back end server. Although processing which forms the back end server 3 and shows a host a virtual disk has been performed in this embodiment, The composition of giving the virtual device driver 31, the memory storage control program 34, and the data transition means 24 to the fiber channel switch 4 like drawing 9 is also possible. Although this embodiment showed the example as for which data transition between disks is made to a host computer transparent, there are various application places. As long as a virtual device driver performs matching with a virtual disk and actual memory storage, for a host computer, data may actually be in any memory storage. With therefore, the memory storage managerial system which can prepare the virtual disk of dynamically required capacity when the necessary minimum virtual disk is defined usually, for example and it is needed and the access frequency of data. It is applicable to the system etc. which move data to a high-speed disk dynamically from a low-speed disk transparent to a host.

[0020]

[Effect of the Invention]According to this invention, all the data manipulation of the data movement between disk units, on-line extension of disk storage capacity, etc. becomes possible transparent entirely to a host computer.

[Translation done.]

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1]It is a block diagram showing the example of composition of the computer system in the embodiment of this invention.
[Drawing 2]It is a table format figure showing the table which the data transition means of the new storage subsystem of this invention uses.
[Drawing 3]It is a flow chart which shows the flow of the data transition processing which the data transition means of this invention performs.
[Drawing 4]It is a flow chart which shows the flow of processing of a data transition means when a write request comes during data transition processing.
[Drawing 5]It is a flow chart which shows the flow of processing of a data transition means when a lead demand comes during data transition processing.
[Drawing 6]In the computer system in the embodiment of this invention, it is a flow chart which shows the flow of processing of a back end server when performing data transition processing to a new storage subsystem from the old storage subsystem.
[Drawing 7]It is a block diagram which realizes the embodiment of this invention and in which showing the example of composition of another computer system.
[Drawing 8]It is a block diagram which realizes the embodiment of this invention and in which showing the example of composition of another computer system.
[Drawing 9]It is a block diagram which realizes the embodiment of this invention and in which showing the example of composition of another computer system.
[Description of Notations]
1 — A host, 2 — The old storage subsystem, 2 — New storage subsystem, 3 — A back end server, 4 — A fiber channel switch, 11 — Operating system, 12 — Application, 13 — An interface, 21 — Disk, 22 [— A virtual device driver, 32 / — An interface, 33 / — An interface, 34 / — A memory storage control program, 41a / — A port, 41b / — A port, 41c / — A port, 41d / — A port, 41e / — Port.] — A controller, 23 — An interface, 24 — A data transition means, 31

[Translation done.]

* NOTICES *

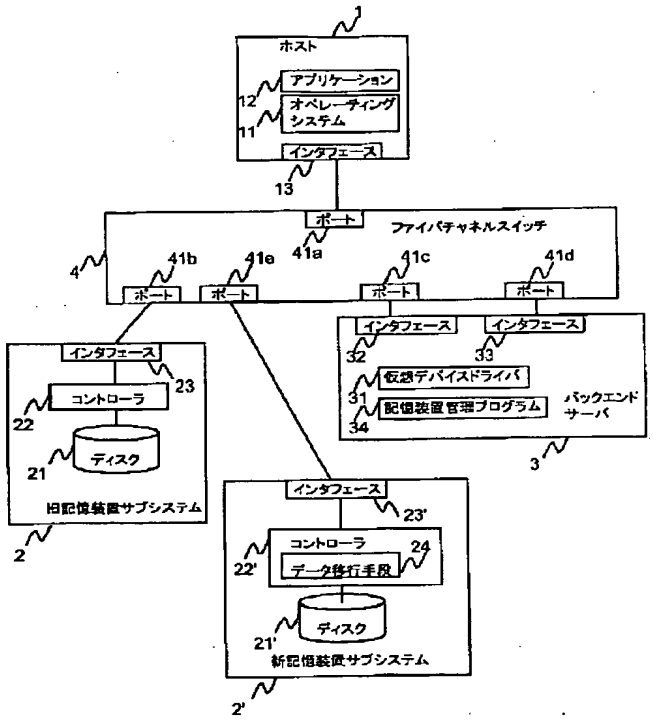
JPO and INPIT are not responsible for any
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

[Drawing 1]

図 1



[Drawing 2]

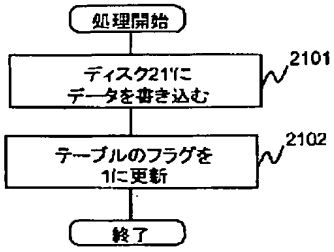
図 2

100 101 102

アドレス	フラグ
LBA0	1
LBA1	1
LBA2	0
LBA3	0
LBA4	1
⋮	⋮
LBAn	0

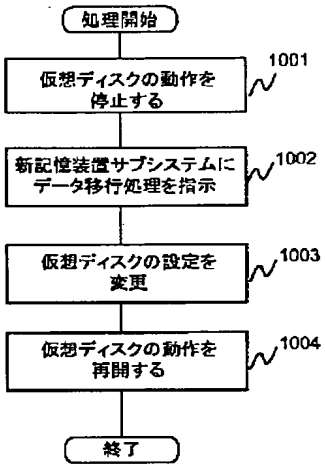
[Drawing 4]

図 4



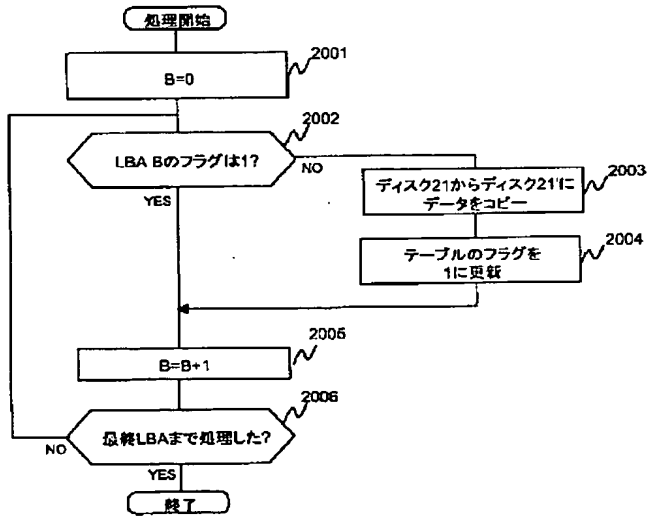
[Drawing 6]

図 6



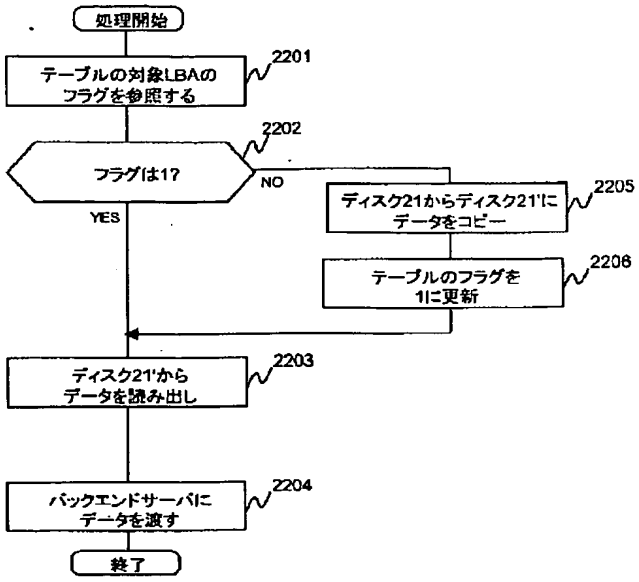
[Drawing 3]

図 3



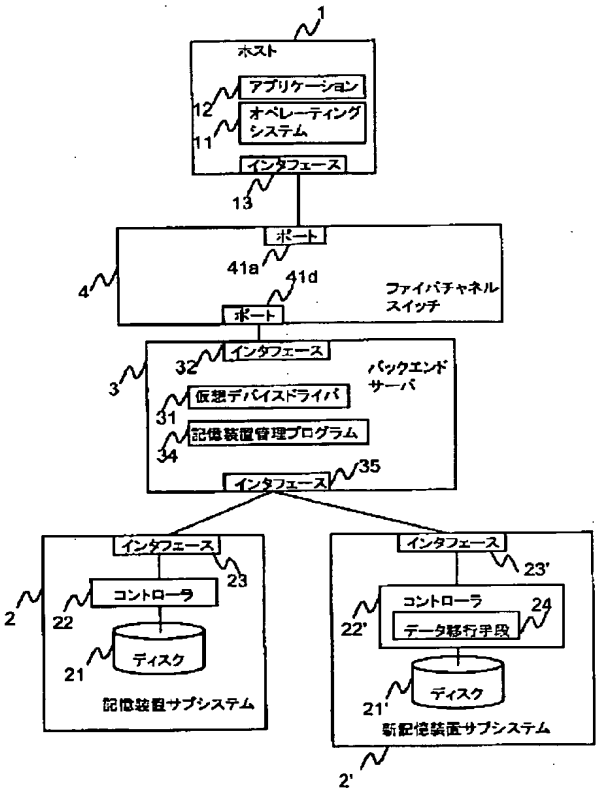
[Drawing 5]

図 5



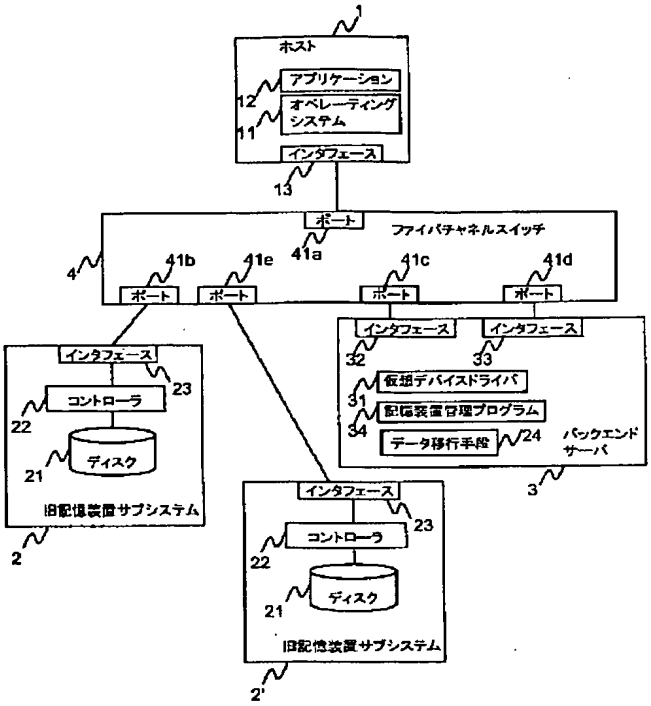
[Drawing 7]

図 7



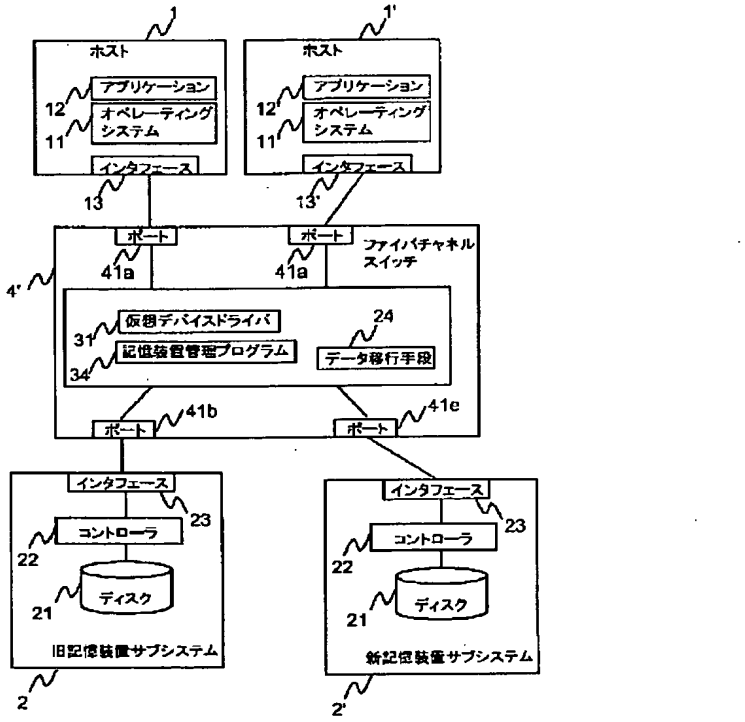
[Drawing 8]

図　8



[Drawing 9]

図　9



[Translation done.]

CLAIMS

[Claim(s)]
[Claim 1]In a computer system which comprised a switch which combines two or more computers, two or more memory storage, and said two or more computers and two or more memory storage mutually, This computer system has a means to provide virtual memory storage to said two or more computers, A computer system which said virtual memory storage is the memory storage corresponding to said at least one memory storage of two or more of said memory storage, and is characterized by a means to provide said virtual memory storage changing said correspondence dynamically.
[Claim 2]A computer system when a means in claim 1 to provide virtual memory storage changes said correspondence dynamically, wherein it does not show that said correspondence changed to said two or more computers.

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2001－331355
（P2001－331355A）
(43)公開日　平成13年11月30日(2001.11.30)

(51)Int.Cl.[7]	識別記号	FI		テーマコード*(参考)
Ｇ０６Ｆ　12/00	５１４	Ｇ０６Ｆ　12/00	５１４	５Ｂ０６５
	５４５		５４５Ａ	５Ｂ０８２
3/06	３０１	3/06	３０１Ｘ	
	３０４		３０４Ｆ	

審査請求　未請求　請求項の数２　ＯＬ　（全７頁）

(21)出願番号　特願2000－152672(P2000－152672)

(22)出願日　平成12年５月18日(2000.5.18)

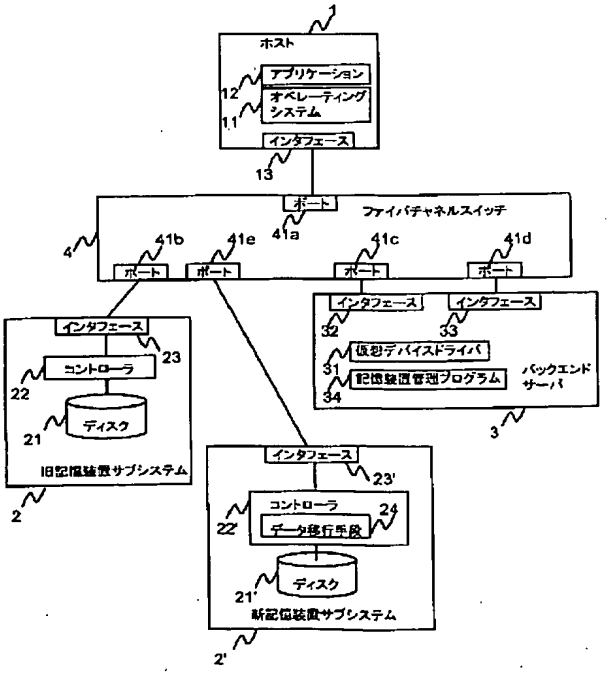
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　北村　学
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内
(72)発明者　荒井　弘治
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内
(74)代理人　100075096
弁理士　作田　康夫

Ｆターム(参考)　5B065 BA01 CE22 EA33 EA35 ZA01
5B082 FA00 HA05

(54)【発明の名称】　計算機システム

(57)【要約】
【課題】複数のホストコンピュータと複数の記憶装置が相互結合された計算機システムにおいて、記憶装置間でのデータの移動をホストコンピュータに対して透過的に実施する。
【解決手段】バックエンドサーバ３はホスト１に対し、仮想ディスクを提供する。仮想ディスクははじめ、旧記憶装置サブシステム２と同じものとしてホスト１に見える。旧記憶装置サブシステム２から新記憶装置サブシステム２にデータを移行する場合、バックエンドサーバ３は最初に新記憶装置サブシステム２にデータ移行処理を指示し、引き続き仮想ディスクの設定を切り替えて新記憶装置サブシステム２に対応させる。この切り替えはホスト１に対して透過的に実施されるため、ホスト１に対して透過的にディスク装置間のデータ移行が可能になる。

図　1



1

【特許請求の範囲】

【請求項１】　複数の計算機と複数の記憶装置と、前記複数の計算機と複数の記憶装置とを相互に結合するスイッチとで構成された計算機システムにおいて、該計算機システムは前記複数の計算機に対し仮想的な記憶装置を提供する手段を有し、前記仮想的な記憶装置は前記複数の記憶装置の少なくとも１つの前記記憶装置に対応する記憶装置であって、前記仮想的な記憶装置を提供する手段は、前記対応を動的に変更することを特徴とする計算機システム。

【請求項２】　請求項１における、仮想的な記憶装置を提供する手段は、前記対応を動的に変更した際に、前記複数の計算機に対しては、前記対応が変化したことを見せないことを特徴とする計算機システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、情報処理システムなどにおける記憶装置システムのデータアクセス方法に係り、特に、記憶装置内のデータ移行方法に関する。

【０００２】

【従来の技術】パソコン、ワークステーション、メインフレームなどの異なるアーキテクチャ、オペレーティングシステムを採用しているプラットフォームと複数の記憶装置とを相互に接続し、いわゆる１つのネットワークにまとめる動きが盛んになっている。これを一般に、複数の計算機をイーサネット（登録商標）（Ｅｔｈｅｒｎｅｔ（登録商標））などのネットワークで接続したLAN(Local Area Network)に対する言葉でSAN (Storage Area Network)と呼ぶ。SANは通常ファイバチャネル(Fibre Channel)という光ケーブルないし銅線の伝送路を用いて計算機と記憶装置を接続する。

【０００３】SANにはいくつかの利点があげられている。まず第１に複数の計算機から記憶装置が共通にアクセスできる環境を提供することである。第２に記憶装置同士も相互接続されることにより記憶装置間でのデータ転送が可能で、これにより、ホスト計算機に負荷をかけることなくバックアップや記憶装置間のデータコピーが実現でき、記憶装置障害時には副系の記憶装置への切り替えが可能となる。第３に、これまで個々の計算機に個々の記憶装置が接続されていたため、記憶装置の管理(装置の状態の監視、設定の変更)は接続されている個々の計算機からしかできなかったものを、特定の計算機から全ての記憶装置の管理を可能にする。また、従来のSCSI(Small Computer System Interface)では最高16台までの機器しか接続できなかったが、ファイバチャネルによって100台以上の機器をオンラインで接続でき、容易な拡張性を得られる。

【０００４】近年、SANを実現するための製品が数多く現れてきているが、実際に上記利点を生かしたものはない。とくに拡張性においては、機器のオンライン接続は

2

物理的に可能になったものの、それを活用する基盤技術が不足している。たとえばSANにおいて、ディスク装置の交換のために新規にディスク装置を増設した場合、機器の増設はオンラインにて実施できるが、そのあとでデータの移動をユーザが明示的に行う必要がある。オンラインの機器増設でユーザがメリットを享受するには、単純なハードウェアの増設だけでなく、ハードウェアの増設に伴いデータ移動などがユーザに対して透過的に実施される必要がある。

【０００５】ディスク装置間の、オンラインのデータの移動に関しては、米国特許5680640号にその例が開示されている。米国特許5680640号はメインフレーム用ディスクを前提としたデータ移行であるが、ディスク装置間を接続する通信線を利用し、ホストとディスク装置間の接続を短時間切断するだけで、あとはユーザに透過的にディスク装置間のデータ移行を可能にしている。

【０００６】

【発明が解決しようとする課題】米国特許5680640号はユーザに対して限りなく透過的にディスク装置間でのデータ移動を可能にしている。ただし、これはメインフレーム用ディスクを前提としたデータ移行方法であり、SANにおいての適用は出来ない。米国特許5680640号では旧ディスク装置を新規ディスク装置に切り替える際、ディスク装置側の設定によって新規ディスクがあたかも旧ディスク装置であるかのようにホスト側に見せかけることが出来る。これはディスク装置のデバイス番号などの設定を操作することで可能である。

【０００７】ただし、SAN、たとえばファイバチャネル環境の場合には、個々のディスクに付与される一意なIDは、ネットワークを構成する機器（ディスク装置、ファイバチャネルスイッチ）同士のネゴシエーションによって決定され、ユーザの設定によって変えられるものではない。米国特許5680640号のデータ移行方法を用いる場合、ホストコンピュータに対して、新規ディスク装置を旧ディスク装置として見せかけることはできず、事実上ユーザに透過的なデータ移行は実現できない。本発明の目的は、ホスト、ユーザに対して透過的で、かつSANの拡張性を生かすことのできるシステムを提供することにある。

【０００８】

【課題を解決するための手段】本発明における計算機システムは、ホスト計算機、バックエンド計算機、複数の記憶装置サブシステムと、ホスト計算機とバックエンド計算機とを接続するスイッチとで構成される。ホスト計算機はバックエンド計算機を介して各記憶装置サブシステムにアクセスするが、バックエンド計算機は、ホスト計算機に対して１つないし複数の仮想的なディスク装置を提供する。ホスト計算機から仮想的なディスク装置にアクセス要求があると、バックエンド計算機では要求のあった仮想的なディスク装置の種類に応じて、実際に接

続されている記憶装置サブシステムに適宜要求を出す。

【０００９】

【発明の実施の形態】図１は、本発明を適用した計算機システムの一実施形態における構成例を示すブロック図である。計算機システムは、ホスト１、旧記憶装置サブシステム２、新記憶装置サブシステム２、バックエンドサーバ３、ファイバチャネルスイッチ４とで構成される。

【００１０】ホスト１はオペレーティングシステム１１、アプリケーション１２、インタフェース１３から構成される。オペレーティングシステム１１、アプリケーション１２は実際にはホスト１上のＣＰＵ、メモリ上で動作するが、これらハードウェアの構成要素については本発明の内容と関係が無いため省略している。実際にはホスト１以外に複数のホストコンピュータがつながる環境が一般的であるが、本発明では簡単のため、ホストコンピュータとしてホスト１のみを記載している。旧記憶装置サブシステム２はディスク２１、コントローラ２２、インタフェース２３とから構成される。ディスク２１は複数の物理ディスクをまとめて１つの論理的なディスク装置に見せかけた論理ドライブであっても、本発明の内容に変わりはない。インタフェース２３はファイバチャネルスイッチ４と接続される。新記憶装置サブシステム２も旧記憶装置サブシステム２と同様、ディスク２１、コントローラ２２、インタフェース２３とから構成される。旧記憶装置サブシステム２と異なる点は、コントローラ２２中にデータ移行手段２４が含まれることである。

【００１１】バックエンドサーバ３は仮想デバイスドライバ３１、インタフェース３２、３３から構成される。仮想デバイスドライバ３１はバックエンドサーバ３上のＣＰＵ、メモリ上で動作するソフトウェアで、ユーザによって外部から設定を変更したりあるいはプログラム自体の入れ替えをすることが可能であるが、ＣＰＵ、メモリなどのハードウェア構成要素に関しては本発明の内容と関係ないため省略している。

【００１２】ファイバチャネルスイッチ４は複数のポート４１ａ、４１ｂ、４１ｃ、４１ｄ、４１ｅ（以下総称してポート４１と略す）から構成され、ホスト１、旧記憶装置サブシステム２、新記憶装置サブシステム２、バックエンドサーバ３を相互に接続するために使用される。ポート４１ａからはいずれもポート４１ｂ、４１ｃ、４１ｄ、４１ｅにアクセスすることが可能である。そのため、ホスト１はポート４１ｂ、４１ｅから直接旧記憶装置サブシステム２や新記憶装置サブシステム２にアクセスすることもできるが、本実施形態においては、基本的にホスト１はすべてバックエンドサーバ３を介して記憶装置サブシステム２にアクセスすることとする。

【００１３】バックエンドサーバ３の役割について説明する。バックエンドサーバ３は仮想デバイスドライバ３１によって、ホスト１から見てあたかも１つないし複数のディスク装置であるかのように見える。本実施形態では、ホスト１がポート４１ｄを介してインタフェース３３を見ると、１つのディスクがつながっているように見えるものとする。以降、このディスクのことを仮想ディスクと呼ぶ。仮想デバイスドライバ３１は、最初はホスト１からは仮想ディスクが旧記憶装置サブシステム２のディスク２１と同じ物に見えるように設定されている。すなわちホスト１が仮想ディスクの論理ブロックアドレス(LBA) ０あるいはLBA１にアクセスすると、仮想デバイスドライバ３１はインタフェース３２、ポート４１ｃを介して、ディスク２１のLBA０あるいはLBA１にアクセスし、結果をインタフェース３３、ポート４１ｄを介してホスト１に返す。本発明の実施形態ではインタフェース３２がディスク２１やディスク２１にアクセスするために使われ、またインタフェース３３がホスト１とのやり取りに使われるようになっているが、１つのインタフェースでこれら２つの役割を行わせることも可能である。また、仮想デバイスドライバ３１の設定を変えることで、仮想ディスクが新記憶装置サブシステム２のディスク２１に見えるようにすることも可能である。設定変更を行った場合、ホストコンピュータから見える仮想ディスクに変化はない。ファイバチャネルインタフェースを持つディスク装置の場合、ホストコンピュータからはポートIDと論理ユニット番号(LUN)で一意にディスク装置が認識できるが、仮想デバイスドライバの設定を変更して仮想ディスクがディスク２１からディスク２１に変更されたとしても、ホスト１に対して見える仮想ディスクのポートIDとLUNは変化せず、ホスト１は実際にアクセスしているディスクが変わったことの認識はない。次に新記憶装置サブシステム２のデータ移行手段２４について説明する。データ移行手段２４は米国特許5680640号に開示されているものと同様の手段を有する。データの移行が指示されると、データ移行手段２４は記憶装置サブシステム２のディスク２１の先頭から順にデータを読み出し、ディスク２１へとデータを書き込む。さらに各ブロックないしは複数ブロック単位に、データの移行が終了したかどうかを記録するテーブルを持ち、移行処理中にリードアクセスがくると、このテーブルを参照し、データ移行が済んでいない領域についてはディスク２１からデータを読み出し、データ移行が済んでいる領域についてはディスク２１のデータを返す。

【００１４】図２はデータ移行手段のもつテーブル100を表したものである。データ移行手段２４ではブロックごとにデータをディスク２１からディスク２１へとコピーしていく。テーブル１００ではそれぞれのアドレス１０１ごとにフラグ１０２を持つ。フラグ１０２が１である場合には、そのアドレスのデータはすでにディスク２１からディスク２１にコピーされたことを示し、０の場合には未コピー　であることを示す。データ移行処理

や、データ移行処理中のリード、ライト処理ではこのテーブル１００を利用する。

【００１５】図３で、データ移行手段２４の行う移行処理の流れを説明する。まずカウンタＢを用意し、初期値を０とする(ステップ２００１)。次にテーブル１００を参照し、LBA Bのフラグ１０２が１かどうかチェックする(ステップ２００２)。フラグが１の場合にはデータ移行が済んでいるため、カウンタＢを１増加する(ステップ２００５)。また、ステップ２００２で、フラグ１０２が０であれば、ディスク２１からディスク２１へとデータをコピーし(ステップ２００３)、テーブル１００の該当するフラグ１０２を１に更新し(ステップ２００４)、ステップ２００５へと進む。ステップ２００６ではディスク２１の最終LBAまで処理したかチェックする。すなわちＢがディスク２１の最終LBAを超えたかどうかチェックし、超えていれば処理を完了し、超えていなければステップ２００２に戻って、処理を繰り返す。

【００１６】次に図４で、データ移行手段が図３のデータ移行処理を行っている間に、上位ホスト、すなわち本実施形態ではバックエンドサーバ３からのライト要求があった場合の処理を説明する。この処理は簡単で、ステップ２１０１でディスク２１にデータを書き込み、ステップ２１０２でテーブル１００の該当するＬＢＡのフラグ１０２を１に更新する。つまり、データ移行処理中にライト処理が行われたＬＢＡについては、ディスク２１からのデータ移行は行われない。

【００１７】図５で、データ移行手段が図３のデータ移行処理を行っている間に、上位ホスト、すなわち本実施形態ではバックエンドサーバ３からのリード要求があった場合の処理を説明する。ステップ２２０１で、テーブル１００内のリード要求のあったＬＢＡについてフラグ１０２を参照する。ステップ２２０２でフラグ１０２が１かどうかチェックして処理を分岐する。フラグが１の場合には、そのＬＢＡについてはディスク２１からのデータ移行が完了しているので、ディスク２１からデータを読み出す(ステップ２２０３)。フラグが０の場合には、そのＬＢＡについてデータ移行が完了していないので、一旦ディスク２１からディスク２１にデータをコピーする(ステップ２２０５)。続いてテーブル１００のフラグ１０２を１に更新して(ステップ２２０６)、ステップ２２０３以降へ進む。ステップ２２０４で読み出したデータをバックエンドサーバ３に渡して処理は完了する。

【００１８】次に、本実施形態のシステムでの、旧記憶装置サブシステム２から新記憶装置サブシステム２へのデータ移行処理について、システム全体の流れを説明していく。データ移行を行う際、ユーザはバックエンドサーバ３に移行を指示する。バックエンドサーバ３から新記憶装置サブシステム２へのデータ移行処理開始の指示は、インタフェース３２を介して新記憶装置サブシステム２に伝えられる。図６はバックエンドサーバ３の処理の流れを説明している。バックエンドサーバ３は移行の指示を受けると、まず、仮想デバイスドライバ３１による仮想ディスクの動作を停止する（ステップ１００１)。これにより、仮想デバイスドライバ３１から旧記憶装置サブシステム２へのアクセスは中断され、仮想デバイスドライバ３１はホスト１から仮想ディスクに対するアクセスコマンドを受け付けても、アクセス中止が解除されるまで応答を返さない。次に記憶装置管理プログラム３４は新記憶装置サブシステム２に対してデータ移行処理の開始を指示する(ステップ１００２)。新記憶装置サブシステム２の行うデータ移行処理については後述する。ステップ１００３では、仮想デバイスドライバ３１がこれまでホスト１に見せていた仮想デバイスの設定を、ディスク２１へのアクセスを行うように変更し、ステップ１００４ではステップ１００１で中止していたアクセスを再開させる。仮想ディスクのアクセスが再開されると、ステップ１００１、ステップ１００２の間にホスト１から仮想ディスクに対してアクセスがきていた場合、そのアクセスは全てディスク２１に対して実施される。

【００１９】また、本実施形態においては、記憶装置サブシステム２とバックエンドサーバ３が直接スイッチにつながった接続形態であったが、図７のように記憶装置サブシステム２がバックエンドサーバ３を介してつながる構成であっても実現は可能である。さらに、新規に増設する記憶装置サブシステム２が図１の例のようにデータ移行手段２４をもたないような場合には、図８のようにバックエンドサーバ３側にデータ移行手段２４を持たせ、バックエンドサーバでデータ移行処理を行わせることで同様のことが実現できる。また、本実施形態ではバックエンドサーバ３を設けてホストに仮想的なディスクを見せる処理を施しているが、図９のように仮想デバイスドライバ３１、記憶装置管理プログラム３４、そしてデータ移行手段２４をファイバチャネルスイッチ４に持たせるという構成も可能である。本実施形態では、ホストコンピュータに透過的にディスク間のデータ移行ができる例を示したが、さまざまな適用先がある。仮想デバイスドライバが仮想ディスクと実際の記憶装置との対応付けを行えば、ホストコンピュータにとっては実際にデータがどの記憶装置にあってもかまわない。そのため、例えば普段は必要最低限の仮想ディスクを定義しておき、必要になった時点で動的に必要な容量の仮想ディスクを用意できるような記憶装置管理システムや、データのアクセス頻度により、ホストに透過的にデータを低速ディスクから動的に高速ディスクに移動するシステムなどに応用できる。

【００２０】

【発明の効果】本発明によれば、ホストコンピュータに対して一切透過的にディスク装置間のデータ移動や、デ

ィスク容量のオンライン拡張など、あらゆるデータ操作が可能となる。

【図面の簡単な説明】

【図１】本発明の実施形態における計算機システムの構成例を示すブロック図である。

【図２】本発明の新記憶装置サブシステムのデータ移行手段の使用するテーブルを示すテーブル構成図である。

【図３】本発明のデータ移行手段が行うデータ移行処理の流れを示すフローチャートである。

【図４】データ移行処理中にライト要求が来たときの、データ移行手段の処理の流れを示すフローチャートである。

【図５】データ移行処理中にリード要求が来たときの、データ移行手段の処理の流れを示すフローチャートである。

【図６】本発明の実施形態における計算機システムにおいて、旧記憶装置サブシステムから新記憶装置サブシステムへのデータ移行処理を行うときの、バックエンドサ＊ーバの処理の流れを示すフローチャートである。

【図７】本発明の実施形態を実現する、別の計算機システムの構成例を示すブロック図である。

【図８】本発明の実施形態を実現する、別の計算機システムの構成例を示すブロック図である。

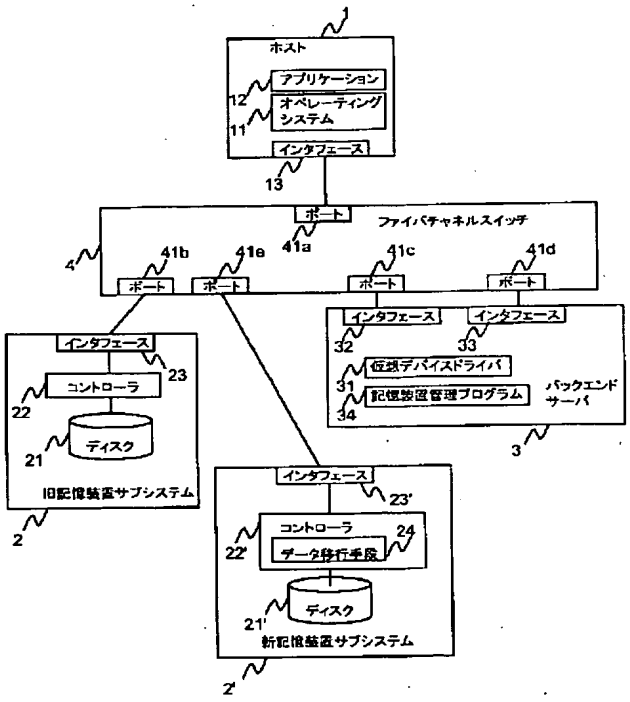
【図９】本発明の実施形態を実現する、別の計算機システムの構成例を示すブロック図である。

【符号の説明】

１…ホスト、２…旧記憶装置サブシステム、２…新記憶装置サブシステム、３…バックエンドサーバ、４…ファイバチャネルスイッチ、１１…オペレーティングシステム、１２…アプリケーション、１３…インタフェース、２１…ディスク、２２…コントローラ、２３…インタフェース、２４…データ移行手段、３１…仮想デバイスドライバ、３２…インタフェース、３３…インタフェース、３４…記憶装置管理プログラム、４１ａ…ポート、４１ｂ…ポート、４１ｃ…ポート、４１ｄ…ポート、４１ｅ…ポート。

【図１】

図 1



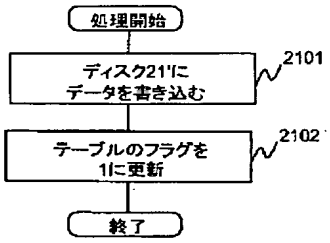
【図２】

図 2

100	101	102
アドレス	フラグ	
LBA0	1	
LBA1	1	
LBA2	0	
LBA3	0	
LBA4	1	
LBAn	0	

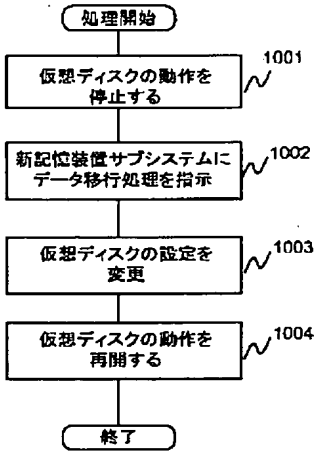
【図４】

図 4



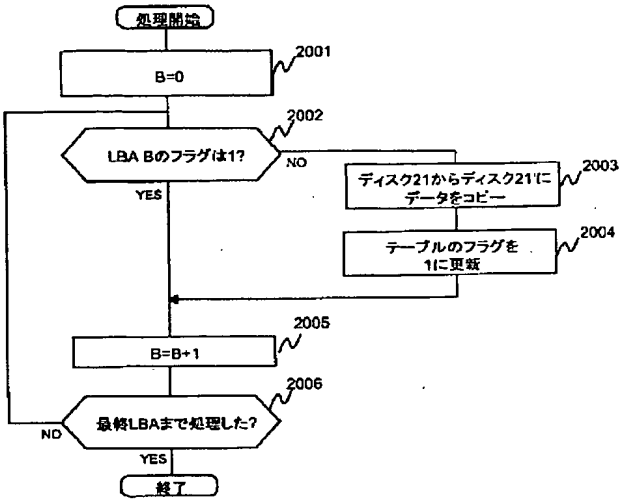
【図６】

図 6



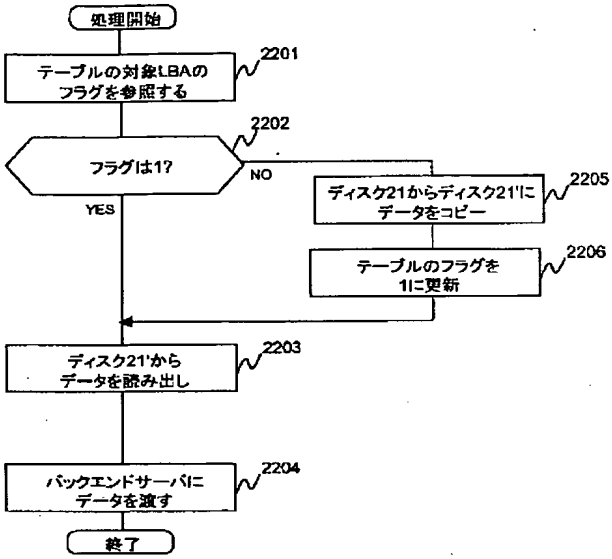
【図３】

図 3



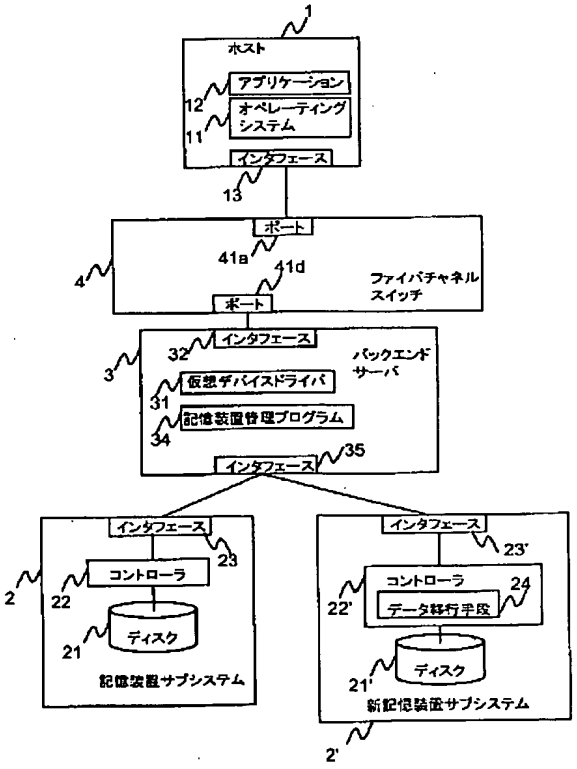
【図５】

図 5



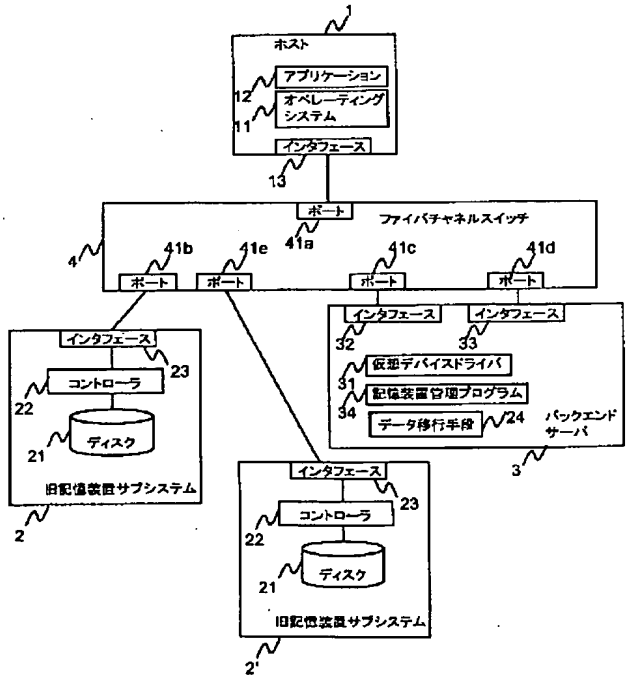
【図７】

図 7



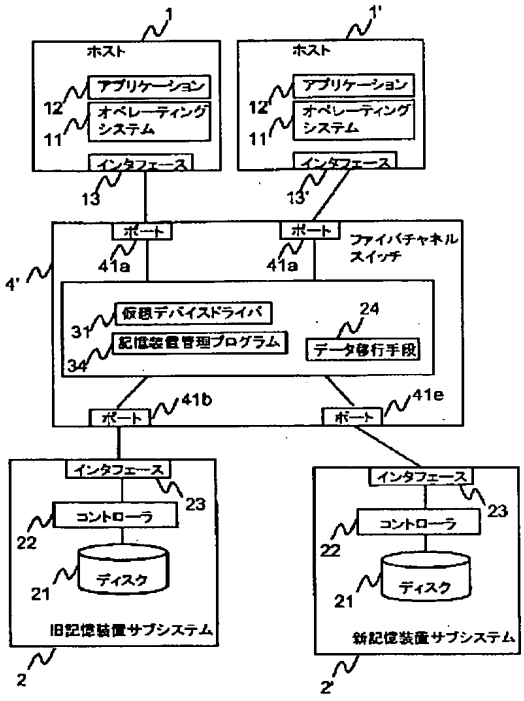
【図８】

図 8



【図９】

図　９


1
ホスト
アプリケーション
12
オペレーティングシステム
11
インタフェース
13
1'
ホスト
アプリケーション
12'
オペレーティングシステム
11'
インタフェース
13'
ポート
41a
ポート
41a
ファイバチャネルスイッチ
4'
仮想デバイスドライバ
31
記憶装置管理プログラム
34
24
データ移行手段
ポート
41b
ポート
41e
インタフェース
23
コントローラ
22
ディスク
21
旧記憶装置サブシステム
2
インタフェース
23
コントローラ
22
ディスク
21
新記憶装置サブシステム
2'